```
#include <stdio.h>
#include <string.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <netdb.h>
// structure definition for designing the packet.
struct frame
{
    int packet[40];
};
// structure definition for accepting the acknowledgement.
struct ack
{
    int acknowledge[40];
};
int main()
{
    int serversocket;
    struct sockaddr_in serveraddr, clientaddr;
    socklen_t len;
    struct frame f1;
    int windowsize, totalpackets, totalframes, i = 0, j = 0, framesend = 0, k,
l, buffer;
    struct ack acknowledgement;
    char req[50];
    serversocket = socket(AF_INET, SOCK_DGRAM, 0);
    bzero((char *)&serveraddr, sizeof(serveraddr));
    serversocket = socket(AF_INET, SOCK_DGRAM, 0);
    bzero((char *)&serveraddr, sizeof(serveraddr));
    serveraddr.sin_family = AF_INET;
    serveraddr.sin_port = htons(5018);
    serveraddr.sin_addr.s_addr = INADDR_ANY;
    bind(serversocket, (struct sockaddr *)&serveraddr, sizeof(serveraddr));
    bzero((char *)&clientaddr, sizeof(clientaddr));
    len = sizeof(clientaddr);
    // connection establishment.
    printf("\nwaiting for client connection");
    recvfrom(serversocket, req, sizeof(req), 0,
             (struct sockaddr *)&clientaddr, &len);
    printf("\nThe client connection obtained\t%s\n", req);
    // sending request for windowsize.
    printf("\nSending request for window size\n");
    sendto(serversocket, "REQUEST FOR WINDOWSIZE", sizeof("REQUEST FOR
WINDOWSIZE"), 0,
           (struct sockaddr *)&clientaddr, sizeof(clientaddr));
    // obtaining windowsize.
    printf("Waiting for the window size\n");
    recvfrom(serversocket, (char *)&windowsize, sizeof(windowsize), 0,
             (struct sockaddr *)&clientaddr, &len);
    printf("\nThe window size obtained as:\t %d \n", windowsize);
    printf("\nObtaining packets from network layer \n");
    printf("\nTotal packets obtained :%d\n", (totalpackets = windowsize * 5));
    printf("\nTotal frames or windows to be transmitted :%d\n", (totalframes =
5));
    // sending details to client.
    printf("\nSending total number of packets \n");
    sendto(serversocket, (char *)&totalpackets, sizeof(totalpackets), 0,
           (struct sockaddr *)&clientaddr, sizeof(clientaddr));
    recvfrom(serversocket, req, sizeof(req), 0,
             (struct sockaddr *)&clientaddr, &len);
    printf("\nSending total number of frames \n");
    sendto(serversocket, (char *)&totalframes, sizeof(totalframes), 0,
           (struct sockaddr *)&clientaddr, sizeof(clientaddr));
    recvfrom(serversocket, req, sizeof(req), 0,
             (struct sockaddr *)&clientaddr, &len);
    printf("\n Press enter to start the process \n");
    fgets(req, 2, stdin);
    // starting the process of sending
    while (i < totalpackets)
    {
        // initialising the transmit buffer.
        bzero((char *)&f1, sizeof(f1));
        printf("\nInitializing the transmit buffer \n");
        printf("\n The frame to be send is %d with packets:", framesend);
        buffer = i;
        j = 0;
        // Builting the frame.
        while (j < windowsize && i < totalpackets)
        {
            printf("%d", i);
            f1.packet[j] = i;
            j++;
            i++;
        }
        printf("sending frame %d\n", framesend);
        // sending the frame.
        sendto(serversocket, (char *)&f1, sizeof(f1), 0,
               (struct sockaddr *)&clientaddr, sizeof(clientaddr));
        // Waiting for the acknowledgement.
        printf("Waiting for the acknowlegment\n");
        recvfrom(serversocket, (char *)&acknowledgement,
sizeof(acknowledgement), 0,
                 (struct sockaddr *)&clientaddr, &len);
        // Checking acknowledgement of each packet.
        j = 0;
        k = 0;
        l = buffer;
        while (j < windowsize && l < totalpackets)
        {
            if (acknowledgement.acknowledge[j] == -1)
            {
                printf("\nnegative acknowledgement received for packet:%d \n",
f1.packet[j]);
                printf("\nRetransmitting from packet:%d \n", f1.packet[j]);
                i = f1.packet[j];
                i = f1.packet[j];
                k = l;
                break;
            }
            j++;
            l++;
        }
        if (k == 0)
        {
            printf("\n Positive acknowledgement received for all packets,within
the frame:%d \n", framesend);
        }
        framesend++;
        printf("\n press enter to proceed \n");
        fgets(req, 2, stdin);
    }
    printf("\nAll frames sends successfully\n Closing connection with the client
\n");
    close(serversocket);
}
```